FLYING LOGIC

User's Guide

version 1.0

Sciral

Sciral
157 N. Glendora Ave. #209
Glendora, CA 91741
FlyingLogic.com

Contents

Getting Started

System Requirements

General	• Free disk space for Flying Logic: 4MB
Windows	• Free disk space for Java: 72MB • Free disk space for Java for 64 bit Intel/AMD: 120MB • Windows XP (SP2) or Windows 2000 (SP4) • Java JRE Version 5.0 update 6 or later (download from java.com/download)
Mac OS X	• Mac OS X 10.4 or later • Java 2 SE 5.0 Release 1 or later (pre-installed, updates downloadable from Software Update control panel or apple.com/java)

Installation

Windows	• Run the installer provided.
Mac OS X	• Drag the application to your Applications Folder.

Editions

Flying Logic is available in several "editions." Certain features are available only in certain editions. All editions can be used to open and review documents created by any other edition.

Reader Edition

The Reader Edition is distributed at no charge, and is used for opening and interactive, read-only review of documents created with any other edition of Flying Logic. Once opened, documents can be exported in the various image formats (PDF, JPEG, PNG), groups may be uncollapsed and collapsed, and confidence values and edge weights may be manipulated, but no changes can be saved.

Student Edition

The Student Edition can be used to create documents that use the basic junctor types (AND, OR) and edge weights (NOT), but advanced operators are not available. Custom domains may be imported into documents, but not created or edited within the Student Edition. Nesting of groups is limited to three levels deep. Document windows and image files contain a watermark.

Personal Edition

The Personal Edition can be used to create documents that use the basic junctor types (AND, OR) and edge weights (NOT), but advanced operators are not available. Custom domains may be imported into documents, but not created or edited within the Personal Edition. There are no restrictions on group nesting, and no watermarks on the screen or image output.

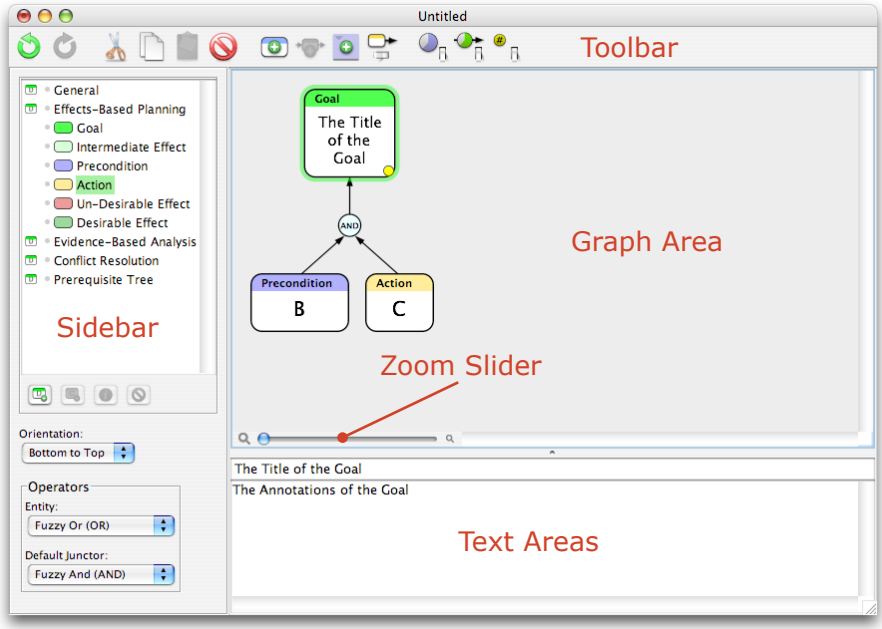
Pro Edition PRO

All features available with no restrictions. Custom domains may be created and exported. Advanced operators are available. The **Edit ▸ Cut/Copy Includes Successors/Predecessors** commands are available. Documents may be exported to Microsoft Project.

In this document, the PRO icon indicates that the described feature is only available in Flying Logic Pro.

The Document Window

Flying Logic can have multiple documents open at one time. Each document window is divided into several functional parts.

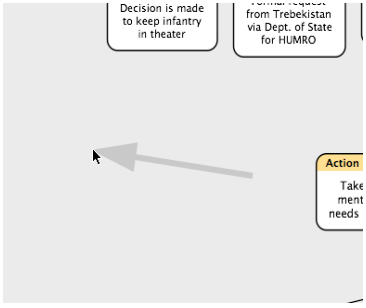


The Graph Area

The Graph Area is the "canvas" where you do most of the work of building your diagram. See the section Constructing Graphs for information on working with the Graph Area.

If the diagram is too big to fit into the window, there are several ways to navigate:

- Use the scroll bars that become active at the bottom and right sides of the Graph Area.
- Use the Zoom Slider to shrink the graph down so more is visible.
- Click and drag in the blank gray background of the Graph Area. This gesture is called "sliding," and causes the view to continuously scroll in the direction of the gray gesture arrow. The scrolling speed is proportional to the distance you drag from the place where you initially click.



Sliding

The Toolbar

The toolbar contains a row of icon buttons that are used to perform common tasks. Each button has a tooltip to remind you of its function that appears when you move the cursor over it. From left to right the icons in the toolbar represent Undo, Redo, Cut, Copy, Paste, Delete, Add Entity, Insert Entity, Add Group, Add Entity as Successor, Show/Hide Confidence Values, Show/Hide Edge Weights, and Show/Hide Note Numbers.

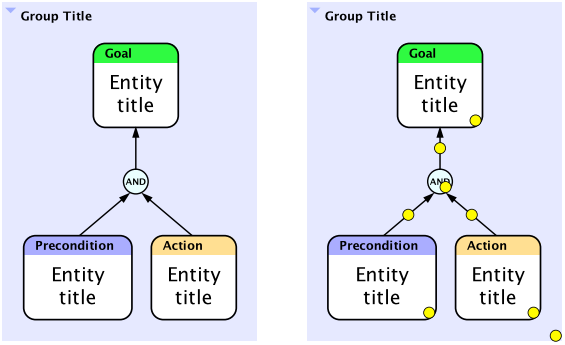
The Sidebar

At the top of the sidebar is the **entity class list**, which includes the kinds of entities ("entity classes" or just "classes") you can put into your diagram. There are four predefined groups ("domains") of classes: the General domain includes two classes: Generic and Note, and the other three include several additional classes each. Which entity classes you use will depend on the needs of your project. For information on using the entity class list to construct your diagram, see Constructing Graphs. For information on creating your own custom classes PRO, see Domains.

At the bottom of the sidebar are several popup menus. The first of these lets you choose the orientation of your diagram. Flying Logic documents all "flow" from their beginning to their end, and this popup lets you determine the direction of this flow. The other popup menus let you select which operators you are using in your document. For more information, see Operators.

The Text Areas

There are two text fields in the Text Areas. The bottom, larger field is used to edit annotations. Annotations can be any length and are intended to be where detailed supporting information and underlying assumptions are recorded. Every element (entity, edge, junctor, group) in the diagram can have its own annotations. To create an annotation or edit an existing one, select the element you wish to annotate, then click in the annotations area. If an element has an annotation, it will appear in the diagram with a small yellow circle on it.



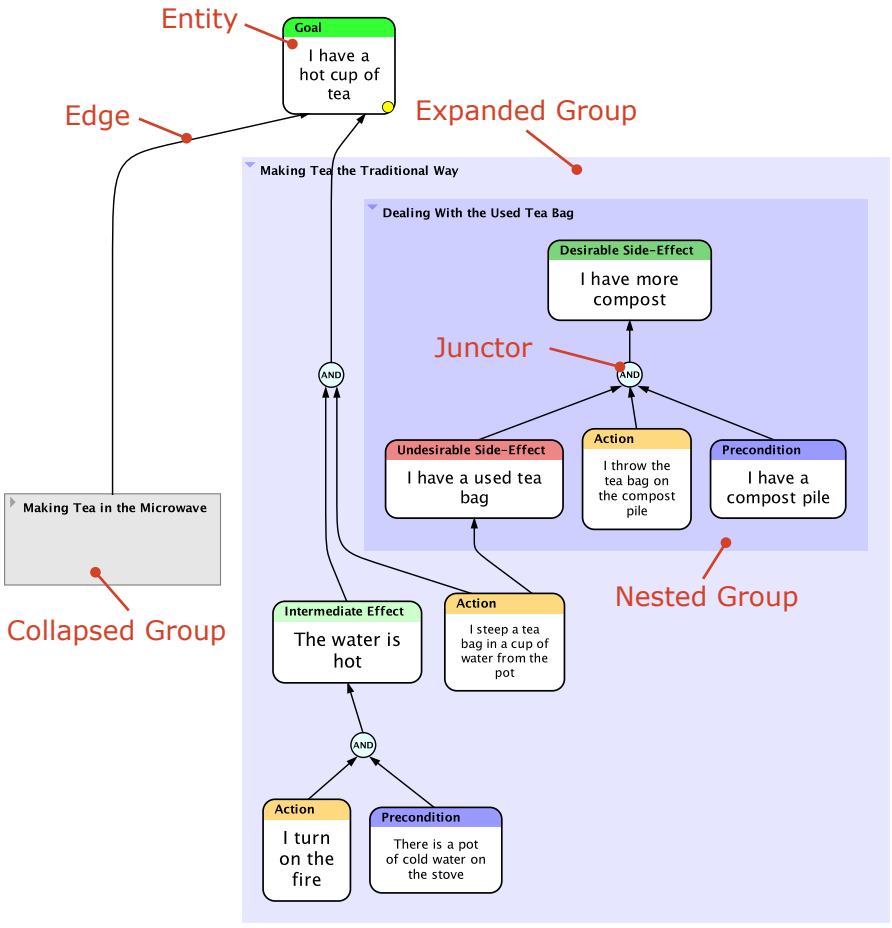
Without Annotations / With Annotations

The top field displays (and can also be used to edit) the title of the currently selected entity or group. See Constructing Graphs for more information about editing titles.

> ***Tip:*** Creating annotations for edges is particularly important, as the edges "hide" your assumptions about *why* you believe one entity causes (or contributes to the cause of) another. Surfacing these assumptions is a critical aspect of using the Thinking Processes.

Constructing Graphs

Flying Logic diagrams are technically known as graphs. A graph is a set of **vertices** (also called *points* or *nodes*) linked by a set of **edges** (also called *lines* or *arrows*.) Graphs can generally be used to represent many things, from street maps to social networks. In Flying Logic, graphs are often used to represent a network of causes and effects called **entities** and the causal relationships between them. The graph can also include **junctors**, which help you control the way the causal relationships combine, and **groups**, which help you organize your documents and manage larger documents.



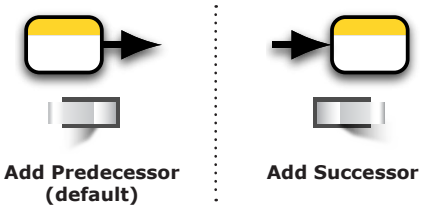
Working With Entities

Entities represent the causes and effects in your diagram. Each entity has a **title**, which is a short descriptive phrase, and can also include a much longer **annotation**. Entities also have a **class**, which is a word or two that describes *what kind* of entity it is, and which appears in a colored bar at the top of the entity. For a description of the classes built into Flying Logic, see the accompanying document, *Thinking with Flying Logic*.

There are several different ways to create new entities within your document. Once you learn them, developing complex diagrams quickly will feel quite natural.

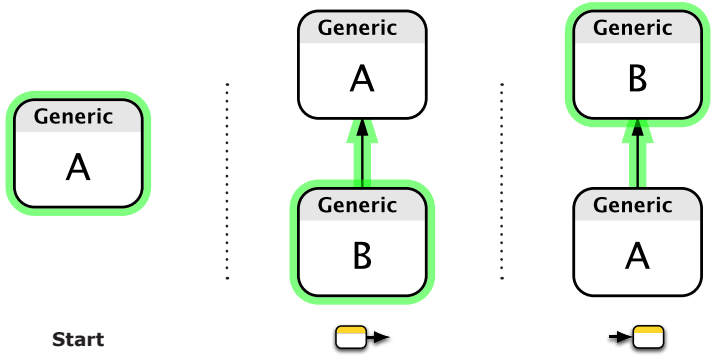
Add Entity as Successor

In the toolbar and **Entity** menu is the **Add Entity as Successor** switch:

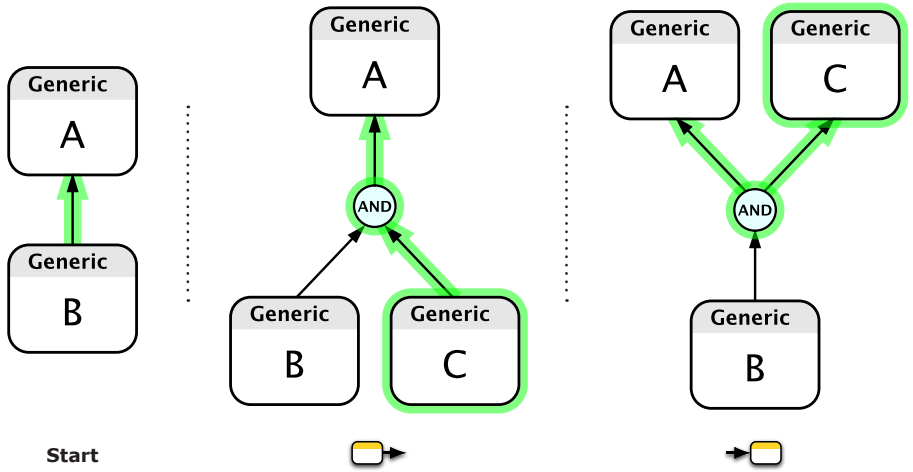


The behavior of several commands and gestures used to add entities to the diagram depends on how this switch is set, and usually you will set it once depending on what sort of diagram you are building. You can also toggle the switch from the keyboard by pressing Command-Option-E (Mac) or Ctrl-Alt-E (Windows).

The commands that use this switch either add a new entity pre-connected to an existing one...



...or they first insert a junctor along the edge, then connect a new entity to it:



In the descriptions that follow, the icon is read as "predecessor or successor," and appears whenever the setting of the **Add Entity as Successor** switch is observed.

Creating Entities With Menu Commands

- Select a class from the class list, then select the **Entity ▸ New Entity** command or click the **New Entity** button in the toolbar. If an open group is also selected in the diagram, the new entity will be added to that group. If an entity or junctor is selected in the diagram, the new entity will be created as a of the selected entity or junctor. If an edge is selected in the diagram, a new junctor will be inserted along the edge, and the new entity will be made a of the new junctor.
- Select a class from the class list and also select an edge in the diagram, then select the **Entity ▸ Insert Entity** command or click the **Insert Entity** button in the toolbar. A new entity will be inserted along the selected edge.
- Right-click (Windows or Mac) or Control-click (Mac) on the gray diagram background or an open group, and select an entity class from the popup menu. A new, unconnected entity will be added to the clicked group or the top level of the document.
- Right-click (Windows or Mac) or Control-click (Mac) on an existing edge or junctor, and select an entity class from the popup menu. If an edge was clicked, a junctor will be inserted along the edge, and the new entity will be added as a of the new junctor.

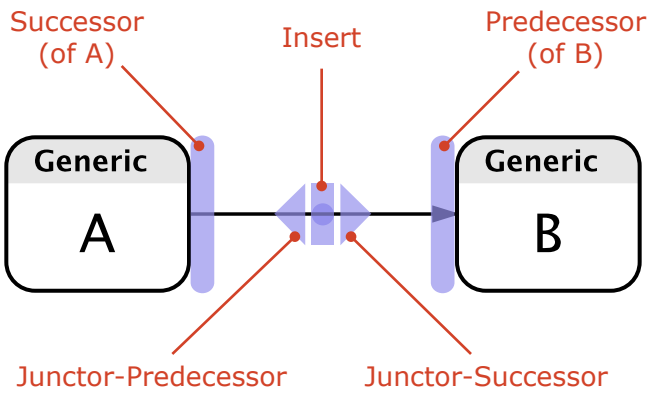
Creating Entities With Drag and Drop

- Drag a class from the class list, and drop it either on an open group, or on the gray diagram background. A new, unconnected entity will be added to the target group, or the top level of the document.
- Drag a class from the class list, and drop it on an existing entity or junctor. The new entity will be created as a  of the target entity or junctor.
- Drag a class from the class list, and drop it directly on an existing edge. A new junctor will be inserted along the edge, and the new entity will be made a  of the new junctor.

Creating Entities With QuickZones

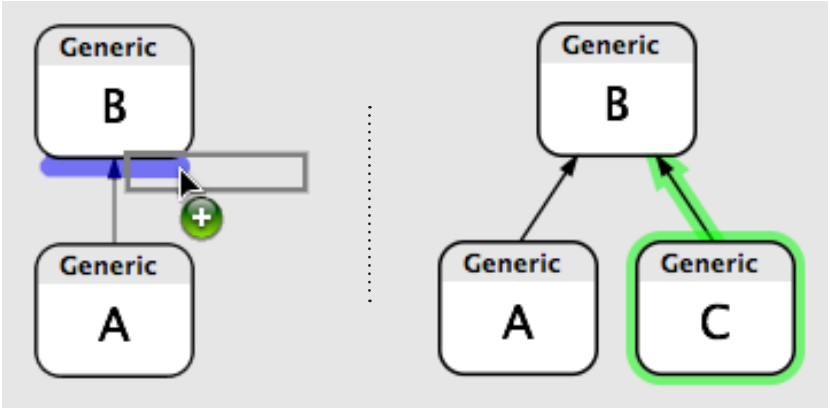
Flying Logic also has special drop zones called QuickZones that allow you to easily place new entities in precise relationships to existing diagram elements without needing to use the **Add Entity as Successor** switch. With a little practice, QuickZones will become your favorite way to construct Flying Logic diagrams.

QuickZones are *near each edge, but not on the edge*, and highlight when you move the cursor over them while dragging a class from the class list.

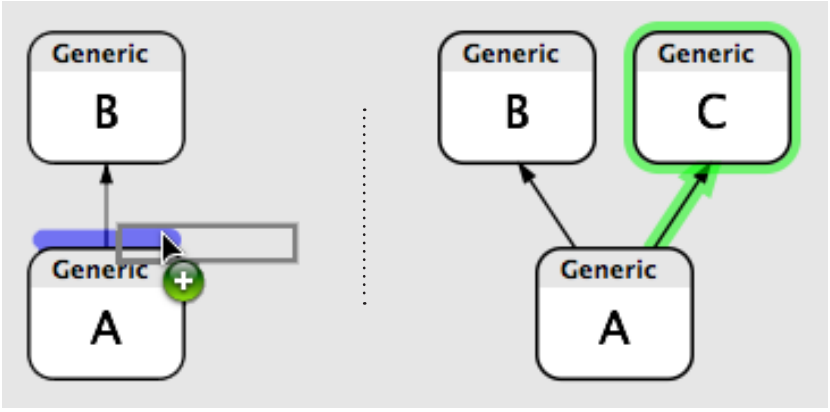


The position of the QuickZones depends on the setting of the diagram orientation popup in the sidebar. In the examples below, the orientation is set to **Bottom to Top**. In this case, the Predecessor QuickZone is underneath entities or junctors. If the orientation were set to **Left to Right**, as in the diagram above, then the Predecessor QuickZone would be to the left of entities or junctors.

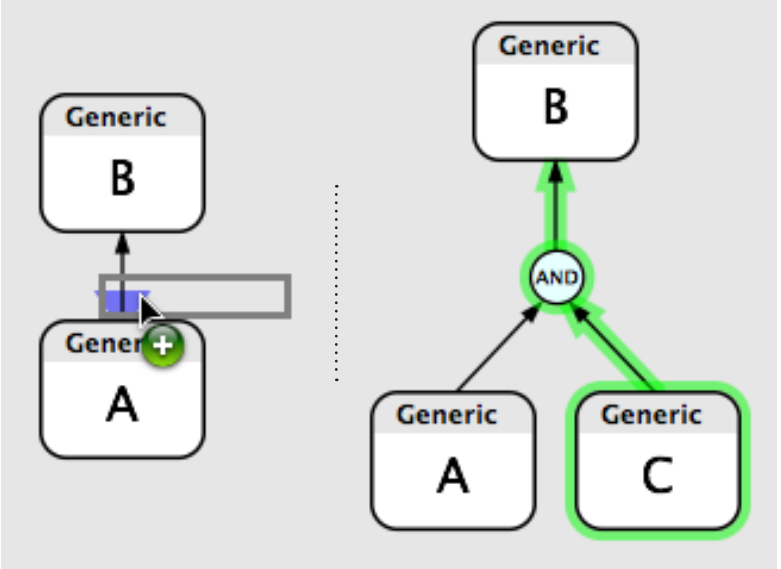
- Drag a class from the class list to the Predecessor QuickZone of an existing entity or junctor:



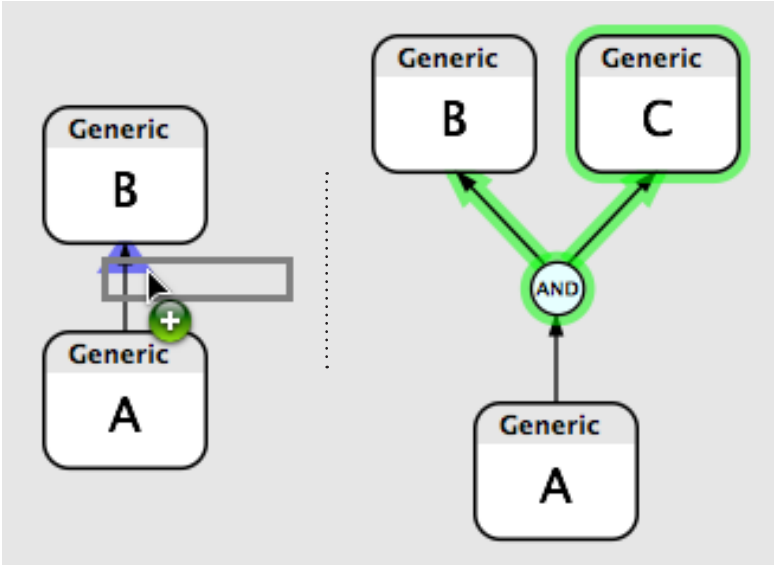
- Drag a class from the class list to the Successor QuickZone of an existing entity or junctor:



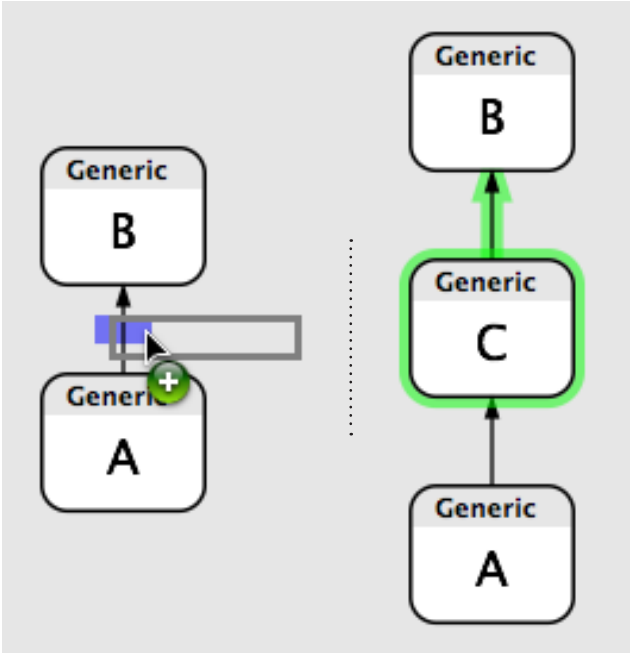
- Drag a class from the class list to the Junctor-Predecessor Quick-Zone of an edge:



- Drag a class from the class list to the Junctor-Successor Quick-Zone of an edge:



- Drag a class from the class list to the Insert QuickZone of an edge:



Changing the Class of an Entity

To change the class of an existing entity, right-click (Windows or Mac) or Control-click (Mac) on the entity and select the desired class from the popup menu.

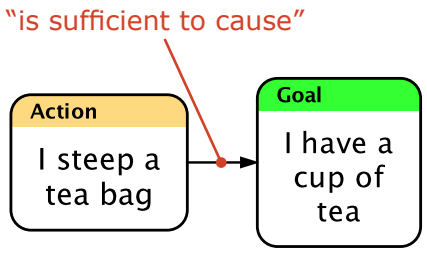
Editing Entity Titles and Annotations

Double-click the entity to begin editing its title, or with an entity selected, press the Tab key. An editor appears over the entity that lets you type or modify the entity title. When you are done editing, press Enter/Return or click outside the editor to commit your changes. You can also press Tab to commit your changes and begin editing the entity's annotation. Finally, you can press Escape to discard your changes.

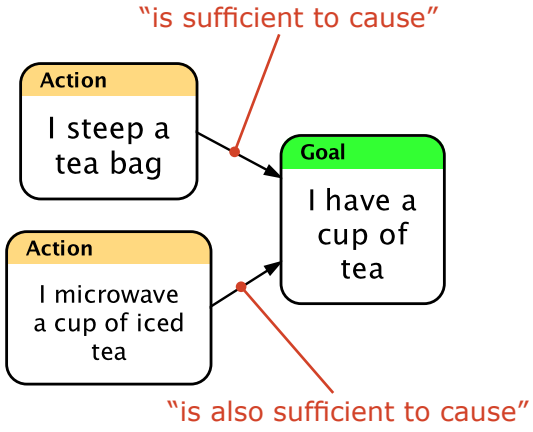
You can directly edit an entity's annotation by selecting the entity then clicking in the annotations area. When you are done editing, commit the change by clicking in the background of the diagram area. To remove an annotation, select and delete all the annotation text.

Working With Edges

In Flying Logic, entities are connected to one another by causal relationships represented by **edges**. Each edge has a direction indicated by its arrowhead. In the way a Flying Logic document is set up by default, each edge represents the principle of *sufficient cause*. In other words, the existence of the state of affairs described by the entity at the tail of the arrow is *sufficient to cause* the state of affairs described by the entity at the head of the arrow. For a discussion of other possible ways to set up the meaning of edges, see Operators.

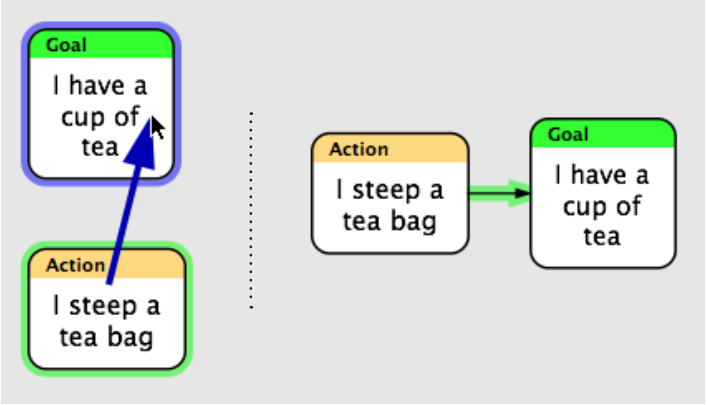


When there are two or more sufficient causes of an entity, then two or more edges point to it. The meaning here is that either OR both of the causes being true makes the effect true.



Creating Edges

To create an edge between entities, click and drag from the cause to the effect. While you drag, a distinctive gesture arrow appears to confirm that you are creating a connection.



Making a Connection

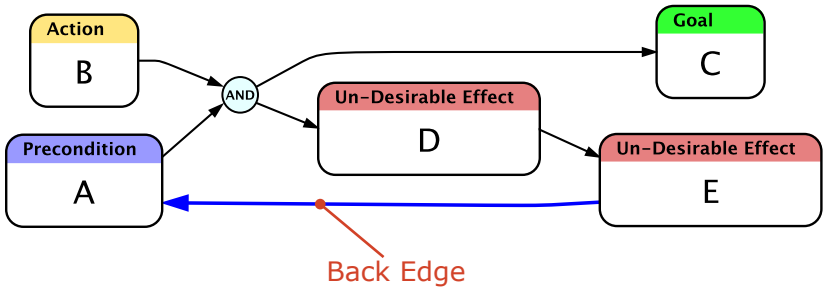
Like other graph elements, edges may have annotations attached to them.

Parallel Edges

Flying Logic disallows *parallel edges*— in other words, more than one edge from a particular cause to a particular effect.

Back Edges

Flying Logic supports **back edges** (also called *cyclic edges* or *loops*), which are edges that (directly or indirectly) make an effect to be its own cause. When a connection is made that requires a loop, a back edge is created automatically. Back edges are thicker than regular edges and are drawn in blue.



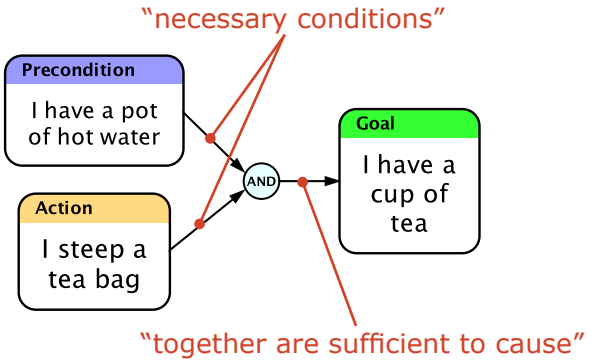
Unlike regular edges, back edges do not participate in calculations, and do not have edge weights. They can be annotated like any other

edge. Supporting back edges allows relationships such as virtuous and vicious circles to be diagrammed, while keeping the basic model a directed acyclic graph.

> ***Tip:*** Your job is to make a diagram that accurately models your problem. Other than a few simple choices, such as whether your diagram flows from bottom-to-top or left-to-right, working out how the various graph elements are laid out is purely Flying Logic's job. Sometimes Flying Logic may surprise you by making layout choices different from those you might have made, but it will often make better choices than most people, especially when the graph becomes complex. Each time you change the structure of the graph, Flying Logic uses animated transitions to show you what has changed. You can control the speed of the animation via the Preferences.

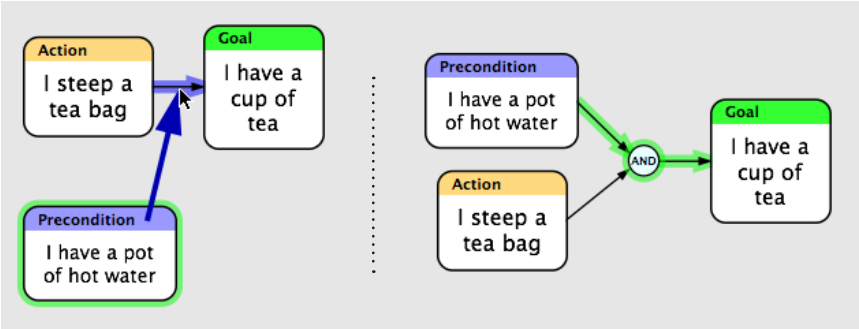
Working With Junctors

Frequently, effects are brought about only by the several causes in combination. **Junctors** are small circles that contain the name of an **operator** such as AND or OR. Since more than one edge entering an entity is, by default, already considered to be an OR operation, junctors are often used to represent AND, which is the idea of necessary condition: that several causes are each necessary but none of them alone is sufficient.



Creating Junctors

As we've seen, creating edges is as simple as dragging from cause to effect. Creating junctors is equally simple: drag from the cause to the edge where the new junctor should appear. You can also right-click (Windows or Mac) or Control-click (Mac) an edge and select an operator type from the Insert Junctor section of the popup menu.



Creating a Junctor

Tip: Any combination of dragging from entities, edges, or junctors to other entities, edges, or junctors will result in a new connection being made, as long as it would not result in a parallel edge. If the start or end (or both) of the drag is an edge, a new junctor will be inserted there and the new edge connected to it.

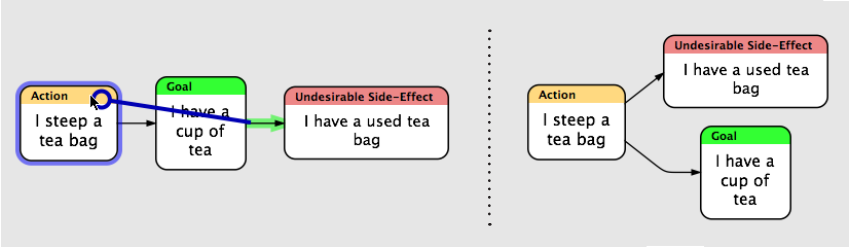
Tip: Preconditions are causes that are outside of your control, while Actions are causes inside your control. For more information about the entity classes used in these examples, see the accompanying document, Thinking with Flying Logic

Entities, Junctors, and Edges

There is a difference between how entities and junctors are related to the edges that connect them. The existence of entities is *independent* of their edges, while the existence of junctors is *dependent* on their edges. That is, an entity can exist whether or not it is connected to any other part of the diagram: it may have edges entering it or leaving it, or both, or it may be unconnected to anything. On the other hand, junctors must always have at least one edge coming in and one edge going out— if the last edge either entering or leaving a junctor is removed, the junctor itself will also be automatically removed.

Moving Edges

As you build your diagram, you will often discover that the head or tail of an edge needs to be moved to a different entity (or junctor). This is easily accomplished by clicking and dragging near the end of the edge you want to move and dropping over the entity (or junctor) where you want to reconnect it. While you drag, a distinctive gesture arrow with a circular head appears to confirm that you are moving an edge. You can also drop over another edge, which will cause a new junctor to be inserted on that edge.



Moving an Edge

Working With Groups

Groups help you organize your diagram and manage the complexity of larger diagrams. Groups appear as shaded rectangles that enclose entities, junctors, and other groups. Like entities, groups may have both titles and annotations, which are edited in exactly the same ways as for entities.

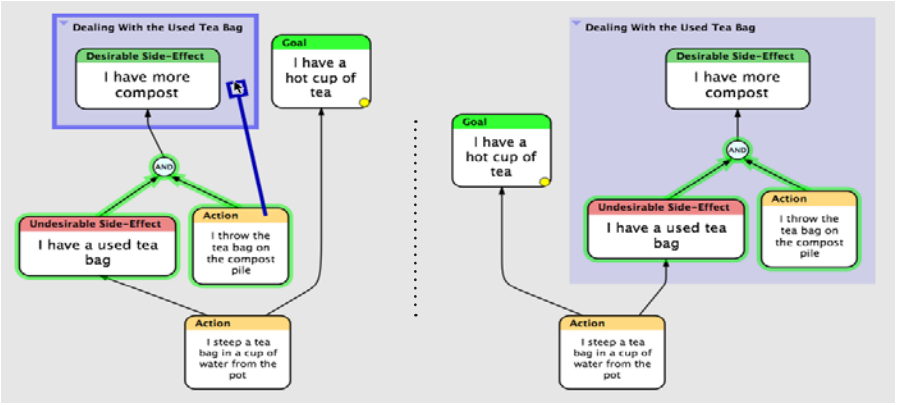
Creating Groups

You can create groups using the **Entity ▸ New Group** command, or by clicking the New Group button on the toolbar.

- If there is no selection, the new group will be created by itself outside all other groups.
- If there is a selection, the new group will enclose every element in the selection.

Arranging Grouped Elements

Individual elements and selected sets of elements can be moved into, out of, and between groups by dragging. First, select the elements you wish to move to a different group. Then drag any of the selected elements into the new group. While you drag, a distinctive gesture arrow with a square head appears to confirm you are rearranging objects within groups. Groups can also be nested within other groups using this technique, and you can also move elements out of any group by dragging to the gray background.

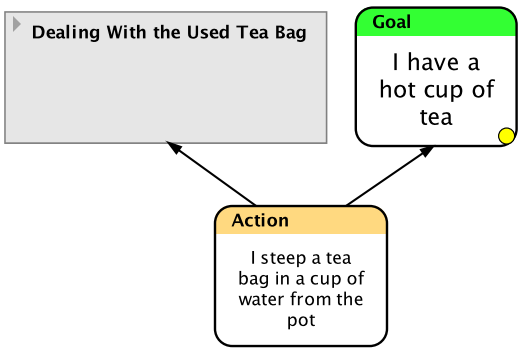


Moving Elements Into a Group

Collapsing and Expanding Groups

Groups can be collapsed to hide their enclosed elements. To collapse a group, click the small triangle that appears in its upper-left corner. Command-click (Mac) or Control-click (Windows) to collapse (or expand) all nested groups within the group as well.

When a group is collapsed, any edges connected to elements both inside and outside the group appear to be attached to the group. Expanding the group will show the actual point of attachment.

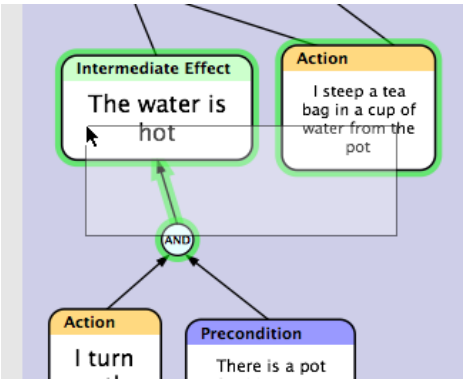


A Collapsed Group

Selections

Flying Logic lets you select several graph elements upon which to perform operations.

- You can select an individual element by clicking it.
- Additional elements can be selected (or deselected) by Shift-clicking.
- **Marquee selection** can be used to select several elements with one gesture. Option-click (Mac) or Control-click (Windows) in the gray background area of the graph and drag the selection rectangle over the elements you wish to select. Additionally, starting marquee selection with Shift-Option-click (Mac) or Shift-Control-click (Windows) will toggle the selected/not-selected state of the objects you drag over. If you begin your marquee selection inside a group, only elements within that group are eligible for selection.



Marquee Selection

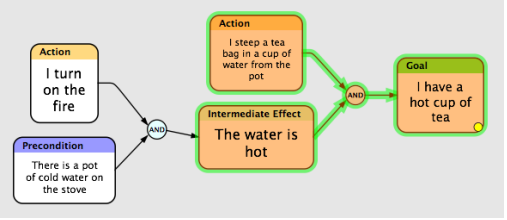
Deleting Selections

All selected elements can be deleted by pressing the Delete key, or by clicking the Delete button in the toolbar. Like all Flying Logic actions, deletion is undoable.

> ***Tip:*** Deleting a group does not automatically delete any enclosed elements (unless they are also explicitly selected.) Instead, the enclosed elements are "promoted" to the next higher group, if any. Deleting a selected group along with all its enclosed elements (including other groups) can be accomplished by pressing Command-Delete (Mac) or Control-Delete (Windows).

Copy and Paste

Selected elements can be cut, copied, and pasted within and between Flying Logic documents. When you make a selection and select the **Edit ▸ Copy** or **Edit ▸ Cut** command, or click the corresponding toolbar icons, the selection becomes shaded orange (for copy) or red (for cut). You can cancel a pending Copy or Cut any time by pressing Escape. Once you have a shaded selection, select the group (or no group at all) in any document you wish to transfer the elements to and select **Edit ▸ Paste** or click the Paste toolbar icon.



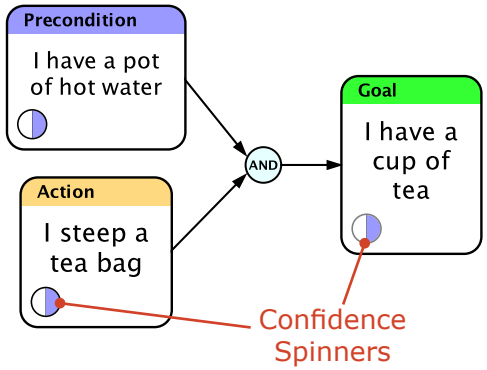
Copied Elements

Graph Logic

Flying Logic provides tools for you to test and work with the logic of your diagrams.

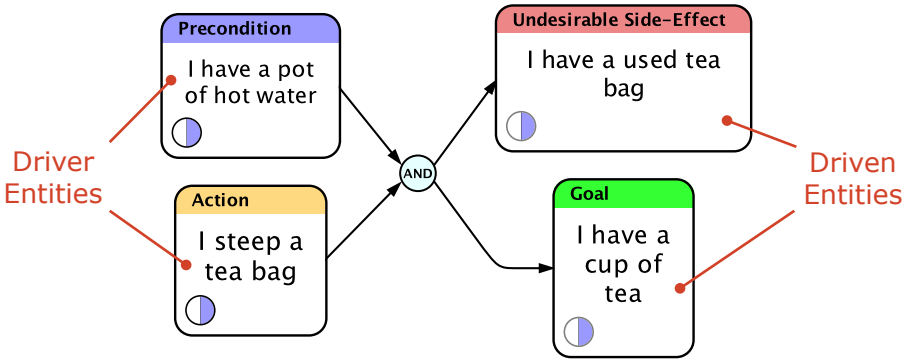
Confidence Spinners

By selecting the **View ▸ Confidence** command or clicking the **Confidence** toolbar icon, each entity reveals a small circle called a **spinner**. Confidence spinners represent a single numeric value that can be thought of as a percentage from 0% to 100% (0.0 to 1.0). You can use Flying Logic's preferences to choose whether spinners are displayed with shading only (no symbol), a numeric value from 0 to 100, or a symbol, which for confidence spinners displays **F** (false) for 0%, **T** (true) for 100%, and shading only for other values.



Setting Confidence Values

Entities that have no incoming edges are **drivers**, and the value of their confidence spinner can be changed directly by clicking and dragging on the spinner. Entities that have one or more incoming edges are **driven**, and their confidence value is purely the result of how their inputs are combined. Spinners of driven entities are bordered by a gray circle, and dragging on them has no effect. For drivers, you can drag right or up to increase the spinner's value, and left or down to decrease it. As you drag, all driven entities affected by your changes are updated simultaneously.



You can also hold down the Shift key while dragging to constrain the spinner's value to increments of 5%, and (while the mouse button is still down) use the arrow keys on your keyboard to "nudge" the spinner's value up or down in 1% increments.

The Meaning of "Confidence"

Confidence is simply a numeric value, but what it *actually* means is subject to the needs of your project and the operators you use to manipulate it. In the way Flying Logic documents are set up by default, the confidence value can be thought of as your level of confidence that the state of affairs described by a given entity holds. If the spinner is 0% shaded then you are expressing absolute certainty that the entity *does not hold* (logical **False**.) If the spinner is 100% shaded then you are expressing absolute certainty that the entity *does* hold (logical **True**.) If the spinner is 50% shaded, then you are expressing a logical **Indeterminacy** which could mean either *no opinion* or *conflicting opinions* about whether the entity holds.

Tip: There is a useful distinction between having *no opinion* and having *conflicting opinions* on a subject, although Flying Logic does not currently support modeling that distinction. In the case of having no opinion, you are asserting *neither* truth nor falsehood. In the case of having conflicting opinions, you are asserting *both* truth and falsehood. There are mathematically valid ways of representing such paraconsistent logic that Flying Logic may support in the future. For typical applications, it is usually most useful to think of confidence values as meaning *no opinion*.

Probability vs. Fuzziness

It is important to understand that in the way Flying Logic documents are set up by default, confidence values represent fuzzy boolean values, not probabilities. The basic distinction to remember is that probabilities deal with the *uncertainty* or *likelihood* of whether something may occur, while fuzzy logic (from which fuzzy booleans derive) deals with *ambiguity* or *vagueness* about what has, or has not, occurred. The following statement is probabilistic:

The water in the pot may get hot.

...while the following statement is fuzzy:

The water in the pot is hot enough.

Combining probabilities is done using multiplication, as in the case of determining the probability of two coin flips both coming up heads (50% × 50% = 25%). To combine fuzzy statements you use fuzzy AND, OR, and NOT operators, which correspond to mathematical minimization, maximization, and complement. ("hot water" AND "steep tea bag" SUFFICIENT FOR "hot tea"). So far we have seen examples of AND and OR. See Edge Weights for a discussion of fuzzy NOT. See Operators for a discussion of how to set up Flying Logic documents for probabilities.

Operators

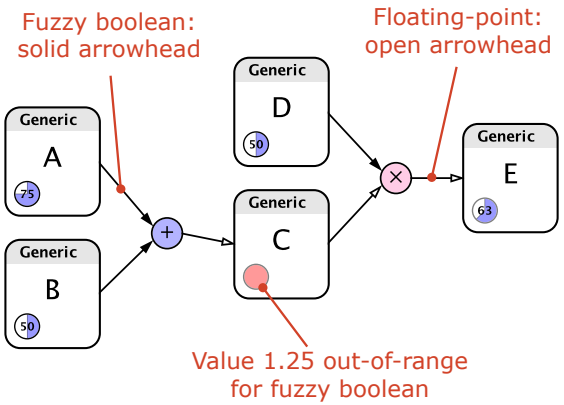
Confidence values flow through the entities, edges, and junctors of a Flying Logic document, being modified along the way by the operators they encounter. This section discusses the operators found on entities and junctors— edges also perform a weighting function discussed in Edge Weights.

Numeric Data Types

Confidence values are represented as one of two data types: *fuzzy boolean* or *floating-point*. The primary difference between the two types is that fuzzy booleans are restricted to the range 0.0 to 1.0, while floating-point values are not. Some of the operators in this section are primarily for use when dealing with fuzzy booleans, some

are primarily for dealing with floating-point values, and some work just as well with either. Generally, if the result of an operation is outside the range of a fuzzy boolean, it will automatically be converted to a floating-point value. Flying Logic displays different kinds of edge arrowheads depending on whether a fuzzy boolean or floating-point value is flowing through it.

Tip: For now, all numeric input and output in Flying Logic is handled by spinners, which can only handle the range 0.0 to 1.0 (for confidence values) or -1.0 to 1.0 (for edge weights.) It is expected that a future version of Flying Logic will incorporate more data types and additional methods for their input and output.



In the above example, the results of the Sum junctor are 1.25, and since this value is out of the fuzzy boolean range, the arrow leaving the Sum junctor has an open arrowhead, indicating a floating-point value. This value cannot be displayed by confidence spinners, and as a result the spinner of entity C turns red. However, the correct value is propagated, as seen from the results of the Product junctor displayed in entity E.

Basic Operators

The Basic Operators are available in all versions of Flying Logic, and (along with Fuzzy NOT provided by edge weights) are sufficient for creating diagrams that support the TOC Thinking Processes.

Fuzzy And (AND)	Returns the minimum of its inputs. Inputs are interpreted as "necessary conditions." Output value is always fuzzy boolean.
Fuzzy Or (OR)	Returns the maximum of its inputs. Inputs are interpreted as "sufficient causes." Output value is always fuzzy boolean.

Advanced Operators PRO

The Advanced Operators are only available in Flying Logic Pro, and are used to support modeling using probabilities and other advanced applications.

Fuzzy Exclusive Or (XOR)	For output to be true, exactly one input must be true. Inputs are interpreted as "sufficient but mutually exclusive causes." Output value is always fuzzy boolean.
Proportion (::)	Treats each input as a "vote" of a strength proportional to the confidence value and the edge weight. Edge weights of zero count as abstentions and do not affect the output, which is different from a simple average where each zero input tends to reduce the output. Output value is fuzzy boolean unless at least one input is floating-point, in which case the output is floating-point.
Sum (+)	Returns the sum of its inputs. Output value is fuzzy boolean unless at least one input is floating-point or the sum is outside the range of a fuzzy boolean, in which case the output is floating-point.
Sum Probabilities (⊕)	Follows the Specific Addition Rule, also called the OR rule. Useful for calculating the probability of two or more independent events causing a particular outcome. For example, the probability of one OR both of two flipped coins coming up heads is 50% ⊕ 50% = 75%. Output value is fuzzy boolean unless at least one input is floating-point, in which case the output is floating-point.
Product (×)	Returns the product of its inputs. Often used to determine the probabilities of two or more independent events occurring together (the Specific Multiplication Rule, also called the AND rule.) For example, the probability of a first AND second coin flip both coming up heads is 50% × 50% = 25%. Output value is fuzzy boolean unless at least one input is floating-point, in which case the output is floating-point.
Reciprocal (1/n)	Returns the reciprocal of its input. Output value is always floating-point. Often used to implement division by way of a / b = a (1 / b). If more than one input is present, returns the reciprocal of the sum of its inputs.
Negate (-n)	Returns the negation of its input. Output value is always floating-point. If more than one input is present, returns the negation of the sum of its inputs.
Complement (1-n)	Returns the complement (1-n) of its input. Output value is fuzzy boolean unless at least one input is floating-point or the sum is outside the range of a fuzzy boolean, in which case the output is floating-point.
Minimum (MIN)	Returns the minimum of its inputs. Output value is fuzzy boolean unless at least one input is floating-point, in which case the output is floating-point.

Maximum (MAX)	Returns the maximum of its inputs. Output value is fuzzy boolean unless at least one input is floating-point, in which case the output is floating-point.
Average (AVG)	Returns the average of its inputs. Output value is fuzzy boolean unless at least one input is floating-point, in which case the output is floating-point.
Distributor (☼)	The distributor behaves exactly the same as the Sum (+) operator, but is intended as a convenience for situations where a single input value is to be distributed to several outputs in a location of the diagram far away from where the value was originally produced.

Junctor Operators

The purpose of junctors is to combine several inputs into an output value using an operator. The name of the operator is displayed on the junctor itself, and is by default Fuzzy And (AND).

PRO Each junctor in a diagram can use a different operator, and the operator that new junctors are created with is set using the Default Junctor Operator popup menu in the sidebar.

PRO To change the operator of an existing junctor, right-click (Windows or Mac) or Control-click (Mac) the junctor and select the desired operator from the popup menu.

Entity Operators

All entities in a diagram have an operator used to combine the confidence values from their incoming edges. By default this operator is Fuzzy Or (OR).

PRO Unlike junctors which can each have a different operator, all entities in a given document have the same operator, which is set using the Entity Operator popup in the sidebar.

Tip: Selecting a new Default Junctor Operator only affects junctors created after the change, but selecting a new Entity Operator immediately changes the operator used by all entities in the diagram.

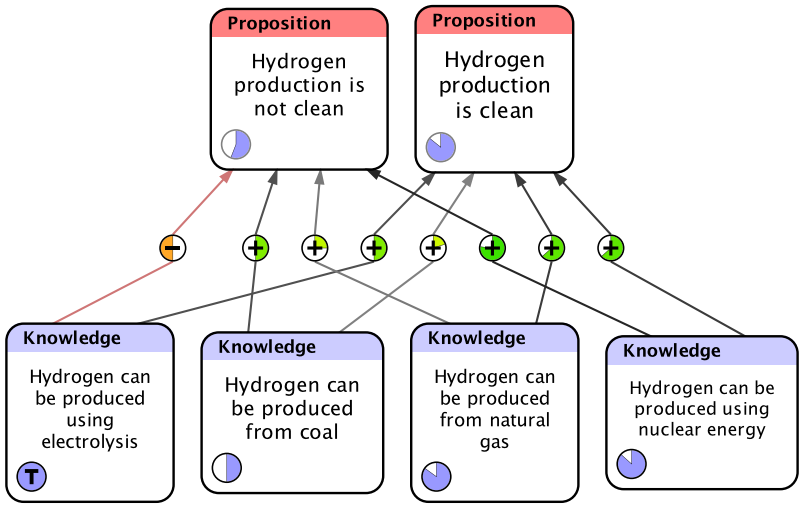
Setting up for Probabilistic Analysis PRO

Usually you will make selections from the Entity Operator and Default Junctor Operator popup menus once when you begin building a diagram. If you are building a diagram using a fuzzy methodology such as Effects-Based Planning, the entity operator Fuzzy Or (OR) and the default junctor operator Fuzzy And (AND) will usually be fine (these are the document defaults.) If you are using a probabilistic methodology such as Evidence-Based Analysis, a common setup is to have Sum Probabilities (⊕) as the entity operator and to have Product (×) as the default junctor operator. This setup is analogous to the use of OR and AND in the default (fuzzy) setup. In cases

where you are evaluating a belief network and wish to use a more intuitive vote-like setup, you will usually set both the entity operator and default junctor operator to Proportion (::).

Edge Weights

Just as each entity carries a confidence value, each edge carries a numerical **edge weight** value. By selecting the **View ▸ Edge Weights** command or clicking the Edge Weights toolbar icon, each edge reveals a spinner that indicates its weight. The weight can be thought of as a percentage from –100% to 100%. Negative weights are represented as red, and positive weights are green. A weight of exactly zero is represented as yellow. The same Preference that selects what kind of symbol to display inside of Confidence spinners also controls the symbol displayed inside edge weight spinners. Another preference lets you decide how the color of the edge itself corresponds to the edge weight. By default, edges with positive weights are black, edges with negative weights are red, and edges with zero weights are gray.



Edge Weights Displayed

In the belief network illustration above, the confidence values of the Knowledge entities represents the strength of belief in each statement, while the edge weights represent the correlation between the knowledge items and the propositions under consideration. The resulting values are combined using the Proportion (::) operator.

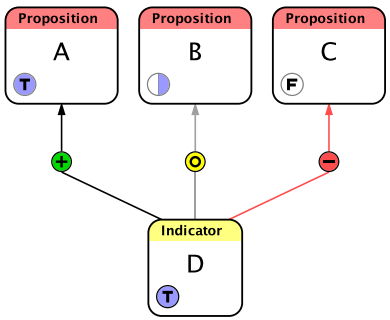
Setting Edge Weight Values

Unlike confidence spinners, there is no "driver/driven" distinction with edge weights— every edge weight may be adjusted independently of all the others. Clicking and dragging on edge weight spinners works the same way as it does for confidence spinners, as does

holding down the Shift key to constrain changes to increments of 5%.

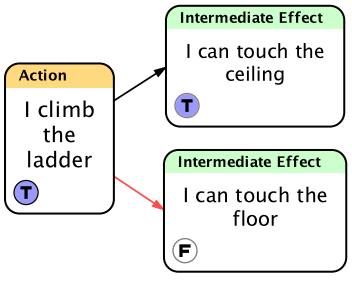
The Edge-Weighting Function

The edge-weighting function is illustrated below. For floating-point values, the result is a simple multiplication. For fuzzy boolean values, the calculation is slightly different. In the case of a simple multiplication, a 100% confidence value (True) multiplied by a 0% edge weight would yield a 0% result (False). What should happen is that a true assertion given no weight should yield an Indeterminate result (neither True nor False). To accomplish this, Flying Logic temporarily converts fuzzy boolean values into the range -100%...100% (like edge weights) multiplies the two values, and then converts the results back into the fuzzy boolean range 0%...100%, which yields the desired result.



Logical Negation

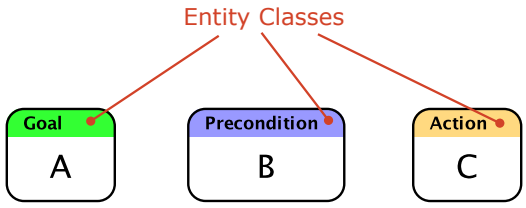
Edge weights are often used in the probabilistic context of Evidence-Based Analysis. But they are also useful in Effects-Based Planning to represent fuzzy NOT (logical negation.) Even without revealing the edge weight spinners, you can right-click (Windows or Mac) or Control-click (Mac) an edge and select a positive (+), neutral (0) or negative (–) edge weight from the popup menu that appears. For fuzzy boolean values, negative edge weights represent the logical negation of the input.



Domains

Classes and Domains

Each entity you create is assigned to an **entity class** (or just *class.*) Entity classes are a purely visual reminder of the "kind" of each entity: Flying Logic does not in any way restrict the kinds of connections an entity may have based on its class. That said, the methodology you choose or problem domain in which you work may place restrictions on what classes of entities may be connected. For instance, in Effects-Based Planning, Precondition and Action entities should always be drivers (have no incoming edges) while Intermediate Effect and Goal entities should always be driven (have at least one incoming edge.)



Working with Classes

To create entities with a chosen class, see Creating Entities.

To select all entities of a chosen class, Shift-double-click the class in the entity class list. To change all selected entities to a chosen class, Alt-double-click (Windows) or Option-double-click (Mac) the class in the entity class list. The last two tasks may be performed in sequence to change all entities of one class to another.

Classes are grouped into **domains**. Domains are used to contain the classes used for creating diagrams in a particular problem domain or using a particular methodology. Double-clicking a domain icon D will collapse or expand that domain. Flying Logic has five built-in domains, which cannot be modified, and which are described in the accompanying document *Thinking with Flying Logic*:

- **General**, which contains two classes, *Generic* and *Note*
- **Effects-Based Planning**
- **Evidence-Based Analysis**
- **Conflict Resolution**
- **Prerequisite Tree**

Creating Custom Classes

PRO Although Flying Logic's built-in classes are useful for many general purposes, in your field you may have particular classes you need to work with. Flying Logic Pro allows you to create the domains and

classes you need, and re-use them in the documents you create. Because the built-in domains and classes are not modifiable, before you can create any custom classes you must create at least one custom domain.

Creating A Custom Domain

1. Select the **Entity ▸ New Domain** command, or click the New Domain button below the class list. The custom domain is created and added to the class list, and the Edit Domain dialog appears.
2. Type a name for the new domain and click the dialog's close box.

Creating A Custom Class

1. In the class list, select one of the custom domains you have created in which you want the custom class to appear, or the existing custom class after which you want the new custom class to be added.
2. Select the **Entity ▸ New Class** command, or click the New Class button below the class list. The custom class is created and added to the selected category, and the Edit Class dialog appears.
3. Type a name for the new class, select a color for the class from the color chooser, and click the dialog's close box. You can now use the new class in your diagram just as you use the built-in classes.

Changing Existing Domains and Classes

To change the name of an existing custom domain or the name or color of an existing custom class, select the domain or class and click the Inspect button at the bottom of the class list.

To delete a custom domain or class, select the domain or class to be deleted and click the delete button at the bottom of the class list.

You can also drag entity classes to re-arrange them within or between domains.

Exporting and Importing Domains

Frequently you will want to re-use the custom domains you create across many documents. Flying Logic Pro supports exporting custom domains, and all editions of Flying Logic support importing custom domains and the classes they contain.

PRO To export a domain, select it in the class list, then select the **Entity ▸ Export Selected Domain** command. The domain file will be saved in the location you specify.

To import a domain, select the **Entity ▸ Import Domain** command, and choose the domain file you wish to import. The category and its classes are added to your document.

You can also create a new, blank document with an exported domain already available by selecting the domain file in the dialog presented by the **File ▸ Open** command, or by double-clicking the domain file in your desktop environment.

> ***Tip:*** You can only export or import an entire domain and all of its classes— you cannot export or import individual classes.

Resolving Conflicts When Importing

Each custom entity class carries a unique, hidden identifier that is created when the class is created, and that never changes even when the name or other attributes of the class changes. These identifiers are exported along with each custom class in the exported category. When a category is imported, a check is done to determine whether any of the identifiers of the classes being imported conflict with the identifiers of the classes already in the document. If any do, it is assumed that an updated version of the category is being imported, and the names and colors of the classes being imported override those already in the document. A notification is provided to you when this occurs, and if the results are not what you expect, you can use the Undo command to revert to the state of your document before the import.

Menus

Application Menu (Mac)

- **About Flying Logic**

Displays version information and credits.

- **Preferences...**

Displays the Preferences dialog.

- **Quit Flying Logic**

Quits Flying Logic. The user is prompted to save changes to any unsaved documents before quitting.

File Menu

- **New Document**

Creates a new, blank document.

- **Open...**

Opens an existing document. New documents can also be opened by dragging a Flying Logic document to the application icon (Windows: in Windows Explorer, Mac: in the Finder or Dock.)

Template files can also be opened, which open as new Untitled documents containing the contents of the template.

Existing exported domain files can also be opened, which causes a new, blank document to be created with the domain automatically imported.

- **Close**

Closes the current document. The user is prompted to save changes to the document if there are any unsaved changes.

- **Save**

Saves the current document. If the document has not been saved before, the user is prompted for a file name and folder to save to.

- **Save As...**

Saves the current document in a new location with a new name.

- **Save as Template...**

Saves a copy of the current document in a new location with a new name, and changes its filename suffix to ".logic-t". When a template file is opened, it opens as a new, Untitled document, but starts with all the content that was saved with the template.

Export Diagram as PDF...

Creates a PDF file with the diagram of the current document. The document is formatted as a single page large enough to contain the entire diagram.

> ***Tip:*** PDF is the most desirable format for publishing and all other applications that do not require a "bitmapped" graphics format (such as JPEG or PNG), as PDF images will scale to any size with no loss of quality. PDF also supports transparency, and will look good when overlaid against backgrounds containing images or textures.

Export Diagram as JPEG...

Creates a JPEG file with the diagram of the current document. The document is formatted as a single image large enough to contain the entire diagram.

> ***Tip:*** JPEG images are compressed, and may show a slight fuzziness around the edges of lines and text. Also, JPEG images do not support transparency, so programs that overlay images on backgrounds (such as presentation software) will show the JPEG as completely obscuring any background. Exporting the image to PDF or PNG is recommended when transparency is desired.

Export Diagram as PNG...

Creates a PNG file with the diagram of the current document. The document is formatted as a single page large enough to contain the entire diagram.

> ***Tip:*** PNG images use a "lossless" form of compression that does not result in the edge-fuzziness associated with JPEG images. Also, PNG images support transparency, and will look good when overlaid against backgrounds containing images or textures.

Export as MS Project Exchange (MPX)...

PRO Exports the current document in MPX format. MPX is a native binary format used to exchange Microsoft Project data. See below for details on the conversion of Flying Logic documents to MS Project documents.

Export as MS Project Data Interchange (XML)...

PRO Exports the current document in MS Project Data Interchange format. MPX is a format based on XML used to exchange Microsoft Project data. See below for details on the conversion of Flying Logic documents to MS Project documents.

When exporting to MS Project, the following conversion rules apply:

- All entities are converted to MS Project Tasks of 1-day duration. The name of the task is the entity's title. The "Text1" field of the task is set to "Entity". The "Text2" field of the task is set to the entity class name. The "Number1" field of the task is set to the confidence value of the entity.
- All junctors are converted to MS Project Milestones. The name of the task is the name of the junctor operator. The "Text1" field of the milestone is set to "Junctor". The "Number1" field of the task is set to the confidence value of the junctor.
- All groups are converted to MS Project summary tasks. The name of the summary task is the title of the group. The "Text1" field of the summary task is set to "Group".
- All edges are converted to "finish-to-start" dependencies with no lag.
- Notes for entities, groups, and junctors become notes on the corresponding tasks. Notes on edges are not preserved, because MS Project does not support notes on task dependencies.

Edit Menu

Undo

Undoes the last action that modified the current document. Virtually every action that modifies Flying Logic documents is undoable, and multiple levels of undo are supported. You can set a Preference as to how many levels of undo to support.

Redo

Redoes the last undone action.

Cut

Works as a typical Cut command for text.

For selections of the diagram, marks the selected elements for transfer to another document or another location in the current document. Selected elements are shaded red. See Copy and Paste.

Copy

Works as a typical Copy command for text.

For selections of the diagram, marks the selected elements for copying to another document or another location in the current document. Selected elements are shaded orange. See Copy and Paste.

Paste

Works as a typical Paste command for text.

For selections of the diagram that have previously been selected for Copy, duplicates the copied elements into the currently selected group, or the top level of the diagram if there is no selection.

For selections of the diagram that have previously been selected for Cut, moves the cut elements into the currently selected group, or the top level of the diagram if there is no selection.

See Copy and Paste.

- **Delete**

Deletes the current selection.

- **Select All**

Selects all elements in the diagram.

- **Deselect**

Deselects all elements in the diagram.

- **Cut/Copy Includes Successors**

PRO If this item is checked, then the Cut or Copy command will not only include selected elements, but also all successor elements to the selected ones. See Copy and Paste.

- **Cut/Copy Includes Predecessors**

PRO If this item is checked, then the Cut or Copy command will not only include selected elements, but also all predecessor elements to the selected ones. See Copy and Paste.

- **Preferences...**

Displays the Preferences dialog.

View Menu

- **Confidence**

Displays or hides the Confidence Spinners.

- **Edge Weights**

Displays or hides the Edge Weights.

- **Note Numbers**

Displays or hides the Note Numbers.

- **Reveal Selection**

Scrolls the current selected elements of the diagram into view, if they are not already visible.

- **Zoom to Fit**

Adjusts the magnification to display the entire diagram.

- **Full Size**

Adjusts the magnification to display the diagram elements at their full size.

Entity Menu

- **New Entity**

Creates a new entity of the class selected in the Entity Class List. See Creating Entities.

- **Insert Entity**

Creates a new entity of the class selected in the Entity Class List and inserts it along the currently selected edge. See Creating Entities.

- **New Group**

Creates a new group that encloses the currently selected elements. See Creating Groups.

- **Reset Confidence**

Resets all confidence values in the entire document to 50%. Useful for restarting demonstrations where all confidence values are expected to begin at 50%.

- **New Domain**

PRO Creates a new domain, represented as a folder in the Entity Class List. See Creating Custom Classes.

- **Delete Domain**

Deletes the currently selected domain, including all classes within the domain. All entities assigned to deleted classes are reassigned to the Generic class.

- **New Class**

PRO Creates a new custom class within the selected domain. See Creating Classes.

- **Delete Class**

Deletes the currently selected class. All entities assigned to the deleted class are reassigned to the Generic class.

- **Import Domain**

Prompts the user for a domain file to import, then adds the domain and its classes to the current document. See Exporting and Importing Domains. A shortcut to create a new document with a domain pre-imported is to simply open the exported domain file using the **File ▸ Open** command or double-click the exported domain file in your desktop environment.

- **Export Selected Domain**

PRO Prompts the user for the name and location of a domain file to save, and then writes the domain selected in the Entity Class List, along with all its classes to the file. See Exporting and Importing Domains.

Window Menu

- **Hide Sidebar**

Hides or displays the sidebar for times when a more presentation-oriented window format is desired.

- **List of Open Documents**

The Window menu also contains a list of currently open documents. The window for an open document may be brought to the front by selecting its name in this menu.

Help Menu

- **Registration...**

This dialog is used for managing your registration information, including your license key.

- **View License Agreement...**

Displays a dialog containing the Flying Logic License Agreement.

- **View Credits...**

Displays a dialog containing credits for Flying Logic.

- **Visit Flying Logic Web Site**

Opens FlyingLogic.com in your web browser, where you can find support for Flying Logic as well as an active community of Flying Logic users.

- **Check for Updates**

Uses your Internet connection to determine whether an updated version of Flying Logic is available for download.

Preferences

The Preferences dialog is displayed using the **Edit ▸ Preferences...** command.

General Options

Auto-Backup on Save

The Auto-Backup on Save checkbox determines whether a backup file of the last saved version is created each time an existing document is re-saved.

Check for Updates Automatically

When selected, Flying Logic will periodically use your Internet connection to check whether an updated version is available for download.

Undo Levels

Flying Logic keeps track of the number of undo levels entered into this box. The default is 100.

Recent Documents

Flying Logic keeps track of this number of recently opened documents in **File ▸ Open Recent**. The default is 20.

Display Options

Each time the diagram changes, an animated transition is shown between the old state and new state. Several aspects of this transition are controllable via preferences.

Fixed-Speed vs. Adaptive Speed

If the Adaptive Animation box is unchecked, each transition will take a fixed amount of time controlled by the Animation Speed (Seconds) slider. If the Adaptive Animation box is checked, each transition will take an amount of time determined from the "visual complexity" of the animation. With adaptive animation, simple changes to the diagram result in quick animations, while complex animations with many objects moving result in slower animations that allow the eye to better track the changes. With adaptive animation, the Speed slider can still be used to control whether the animations happen faster or slower.

Turning Off the Animation

If the Adaptive Animation box is unchecked and animation speed is set to zero seconds, the animation does not take place, and layout changes to the diagram appear instantly.

Animation Style

The animation style radio buttons control how Flying Logic chooses which frames of the animation to generate. If Fixed Frame Rate is selected, then Flying Logic will choose a number of frames over which to perform the animation based on the animation speed (at a nominal 30 frames-per-second) and will draw every one of those frames. This option may look better on slower hardware. The preferred, default style is Fixed Time, where Flying Logic dynamically chooses which frames to draw based on the amount of time the last frame took to draw. This results in animations running at what often feels like a steadier pace, and is particularly suitable to faster hardware. The rule of thumb is to try both options and keep which feels best to you.

Edge Colors

The color of edges in your diagrams depends on this setting. The default is **Red..Gray..Black**, which means that edges with negative weights are drawn in red, edges with neutral settings are drawn in gray, and edges with positive weights are drawn in black. The other options are **Red..Yellow..Black** and **Red..Yellow..Green**.

Spinner Display

The Spinner Display preference determines what symbols are drawn inside of spinners.

	None	Symbol	Numeric
Confidence Spinner	*nothing*	0%: **F** 100%: **T** other values: *nothing*	**0..100**
Edge Weight Spinner	*nothing*	-100%..-1%: **–** 0%: **O** 1%..100%: **+**	**-100..100**

Default Orientation

The Default Orientation setting controls the orientation (flow direction) of newly-created documents. Any document can have its orientation changed using the Orientation popup in its sidebar.